UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/510,203	02/22/2000	Carol A. Fields	X-560 US	2126

24309 7590 05/19/2003

XILINX, INC
ATTN: LEGAL DEPARTMENT
2100 LOGIC DR
SAN JOSE, CA 95124

EXAMINER	
SHARON, AYAL I	

ART UNIT	PAPER NUMBER
2123	3

DATE MAILED: 05/19/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 07-01)

Office Action Summary	Application No. 09/510,203	Applicant(s) FIELDS ET AL.	
	Examiner Ayal I Sharon	Art Unit 2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 February 2000* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *09 March 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Introduction

1. Claims 1-19 of U.S. Application 09/510,203 filed on 2/22/00 are presented for examination.

Claim Interpretations

2. Examiner interprets "design module" according to Applicants' definition in the specification (p.1, lines 18-22): "System-level integration relies on reuse of previously created designs, either from within an enterprise or from a commercial provider. The engineering community sometimes refers to these previously created designs as 'design modules', 'cores', or 'IP' (intellectual property)."
3. Examiner interprets "functional design element" according to Applicant's embodiments in the specification (p.6, lines 31-33): "For example, HDL design constructs are traced from high-level design to design elements. In schematic C, any changes made by a translation tool are tracked." The most basic design elements in HDL, for example, are lines, signals, and AND/OR/NOT gates. However, these can be used to build successively higher-levels of design elements such as flip-flops, memory arrays, adders, multiplexers, etc., all the way up to Memory Caches and Arithmetic Logic Units (ALUs), etc. The broadest reasonable interpretation includes all of these embodiments.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

5. The prior art used for these rejections is as follows:
6. Fields et al., U.S. Patent 6,223,326. (Henceforth referred to as "**Fields_1**").
7. Aleksic et al., U.S. Patent 5,995,736. (Henceforth referred to as "**Aleksic**").
8. The claim rejections are hereby summarized for Applicant's convenience. The detailed rejections follow.
9. **Claims 1-9, 11-14, and 16-19 are rejected under 35 U.S.C. 102(e) as being anticipated by Fields_1.**

The applied reference has a common assignee with the instant application. Based upon the earlier effective U.S. filing date of the reference, it constitutes prior art under 35 U.S.C. 102(e). This rejection under 35 U.S.C. 102(e) might be overcome either by a showing under 37 CFR 1.132 that any invention disclosed but not claimed in the reference was derived from the inventor of this application and is thus not the invention "by another," or by an appropriate showing under 37 CFR 1.131.

10. **Claims 1-19 are rejected under 35 U.S.C. 102(e) as being anticipated by Aleksic et al.**

11. In regards to Claim 1, Fields_1 teaches the following limitations:

> 1. A computer-implemented method for developing a reusable electronic circuit design module, wherein the design.module is comprised of one or more functional design elements comprising the design module, comprising:
>
> entering the functional design elements into a database;
>
> (Fields_1, especially: Fig.1, Item 108 and associated text)
>
> entering documentation elements into the database;
>
> (Fields_1, especially: Fig.1, Item 108 and associated text)
>
> linking the functional design elements with selected ones of the documentation elements;
>
> (Fields_1, especially: Fig.1, Item 106 and associated text)
>
> simulating a testbench with the design module, whereby simulation results are generated;
>
> (Fields_1, especially: Fig.1, Items 110 and associated text)
>
> storing the simulation results in the database; and
>
> (Fields_1, especially: Fig.1, Items 104 and associated text)
>
> linking the simulation results with the functional design elements.
>
> (Fields_1, especially: Fig.3, Items 306-318 and associated text)

12. In regards to Claim 2, Claim 1 is rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

> 2. The method of claim 1, further comprising:
> translating the functional design elements into a netlist; and
> (Fields_1, especially: Fig.1, Item 102 and associated text)
>
> linking elements of the netlist with selected ones of the functional design elements.
> (Fields_1, especially: Fig.1, Item 108 and associated text)

13. In regards to Claim 3, Claim 2 is rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

> 3. The method of claim 2, further comprising:
> translating the functional design elements into a physical implementation; and
> (Fields_1, especially: Fig.1, Item 102 and associated text)
>
> linking elements of the physical implementation with selected ones of the functional design elements.
> (Fields_1, especially: Fig.1, Item 108 and associated text)

14. In regards to Claim 4, Claim 1 is rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

> 4. The method of claim 1, further comprising:
> entering simulation elements in the database; and
> (Fields_1, especially: Fig.1, Item 104 and associated text)
>
> linking the simulation elements to associated ones of the design elements.
> (Fields_1, especially: Fig.1, Items 104 and 108, and associated text)

15. In regards to Claim 5, Claims 1 and 4 are rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

> 5. The method of claim 4, further comprising:
> entering documentation for a design script in the database; and
> (Fields_1, especially: Fig.1, Items 106 and 108, and associated text)

linking the documentation of the design script to the design elements comprising the design module.
(Fields_1, especially: Fig.1, Items 106 and 108, and associated text)

16. In regards to Claim 6, Claims 1 and 4 are rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

6. The method of claim 4, further comprising:
entering documentation for the simulation elements in the database; and
(Fields_1, especially: Fig.1, Item 108, and associated text)

linking the documentation for the simulation elements with associated ones of the simulation elements.
(Fields_1, especially: Fig.1, Item 108, and associated text)

17. In regards to Claim 7, Claims 1, 4 and 6 are rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

7. The method of claim 6, further comprising:
inspecting the functional design elements and simulation elements for associated documentation; and
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

reporting documentation deficiencies in association with the functional design elements and simulation design elements.
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

18. In regards to Claim 8, Claim 1 is rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

8. The method of claim 1, further comprising:
inspecting the functional design elements for associated documentation; and
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

reporting documentation deficiencies in association with the functional design elements.
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

19. In regards to Claim 9, Claim 1 is rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

9. The method of claim 1, further comprising:
inspecting the functional design elements for undesirable design characteristics; and
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

reporting the undesirable design characteristics found in the functional design elements.
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

20. In regards to Claim 11, Claims 1 and 9 are rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

11. The method of claim 9, further comprising:
inspecting the functional design elements for adherence to predefined design rules; and
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

reporting violations of the design rules.
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

21. In regards to Claim 12, Claims 1, 9 and 11 are rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

12. The method of claim 11, further comprising providing assistance in specifying the design rules for the functional design elements
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

22. In regards to Claim 13, Claims 1 and 9 are rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

13. The method of claim 9, further comprising:
monitoring changes made to the functional design elements; and
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

indicating which of the functional design elements are dependent on the changes.
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

23. In regards to Claim 14, Claim 1 is rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

14. The method of claim 1, further comprising:

translating the functional design elements into a physical implementation; and
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

linking elements of the physical implementation with selected ones of the functional design elements.
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

24. In regards to Claim 16, Claim 1 is rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

16. The method of claim 1, further comprising displaying the functional design elements linked to errors in the simulation results.
(Fields_1, especially: Fig.1, Item 106-110, and associated text and Fig.3, Items 306-318 and associated text)

25. In regards to Claim 17, Claims 1 and 16 are rejected over Fields_1 as described above. In addition, Fields_1 teaches the following limitations:

17. The method of claim 16, further comprising displaying documentation elements associated with errors in the simulation results.
(Fields_1, especially: Fig.1, Item 106-110, and associated text and Fig.3, Items 306-318 and associated text)

26. In regards to Claim 18, Fields_1 teaches the following limitations:

18. An apparatus for developing a reusable electronic circuit design module, wherein the design module is comprised of one or more functional design elements comprising the design module, comprising:

means for entering the functional design elements into a database;
(Fields_1, especially: Fig.1, Item 108 and associated text)

means for entering documentation elements into the database;
(Fields_1, especially: Fig.1, Item 108 and associated text)

means for linking the functional design elements with selected ones of the documentation elements;
(Fields_1, especially: Fig.1, Item 106 and associated text)

means for simulating a testbench with the design module, whereby simulation results are generated;
(Fields_1, especially: Fig.1, Items 110 and associated text)

means for storing the simulation results in the. database; and
(Fields_1, especially: Fig.1, Items 104 and associated text)

means for linking the simulation results with the functional design elements.
(Fields_1, especially: Fig.3, Items 306-318 and associated text)

27. In regards to Claim 19, Fields_1 teaches the following limitations:

19. A system for developing a reusable electronic circuit design module, wherein the design module is comprised of one or more functional design elements comprising the design module, comprising:
a database arranged for storage of the design elements and documentation elements;
(Fields_1, especially: Fig.1, Item 108 and associated text)

a design inspector coupled to the database, the design inspector configured and arranged to link the functional design elements with selected ones of the documentation elements;
(Fields_1, especially: Fig.1, Item 106 and associated text)

a debugging-support module coupled to the simulator and to the database, the debugging-support module configured and arranged to generate a netlist from the design module, wherein the netlist is suitable for simulation;
(Fields_1, especially: Fig.1, Items 110 and associated text)

a functional simulator coupled to the debugging-support module, the simulator configured and arranged to simulate a testbench with the design module, whereby simulation results are generated; and
(Fields_1, especially: Fig.1, Items 104 and associated text)

wherein the debugging-support module is further configured and arranged to store the simulation results in the database and link the simulation results with the functional design elements.
(Fields_1, especially: Fig.3, Items 306-318 and associated text)

28. In regards to Claim 1, Aleksic teaches the following limitations:

1. A computer-implemented method for developing a reusable electronic circuit design module, wherein the design.module is comprised of one or more functional design elements comprising the design module, comprising:
entering the functional design elements into a

database;
(Aleksic, especially: Fig.2, Item 38 and associated text)

entering documentation elements into the database;
(Aleksic, especially: Fig.2, Item 46 and associated text)

linking the functional design elements with selected ones of the documentation elements;
(Aleksic, especially: Fig.3, Item 36 and associated text)

simulating a testbench with the design module, whereby simulation results are generated;
(Aleksic, especially: Fig.5, Items 92-98 and associated text)

storing the simulation results in the database; and
(Aleksic, especially: Fig.5, Items 92-102 and associated text)

linking the simulation results with the functional design elements.
(Aleksic, especially: Fig.3, Item 36, Fig.5, Items 92-102 and associated text)

29. In regards to Claim 2, Claim 1 is rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

2. The method of claim 1, further comprising:
translating the functional design elements into a netlist; and
(Aleksic, especially: Fig.4, Items 38, 40, and 49 and associated text)

linking elements of the netlist with selected ones of the functional design elements.
(Aleksic, especially: Fig.4, Items 38, 40, and 49 and associated text)

30. In regards to Claim 3, Claim 2 is rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

3. The method of claim 2, further comprising:
translating the functional design elements into a physical implementation; and
(Aleksic, especially: Fig.5 and associated text)

linking elements of the physical implementation with selected ones of the functional design elements.
(Aleksic, especially: Fig.5 and associated text)

31. In regards to Claim 4, Claim 1 is rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

> 4. The method of claim 1, further comprising:
> entering simulation elements in the database; and
> (Aleksic, especially: Fig.5 and associated text)
>
> linking the simulation elements to associated ones of the design elements.
> (Aleksic, especially: Fig.5 and associated text)

32. In regards to Claim 5, Claims 1 and 4 are rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

> 5. The method of claim 4, further comprising:
> entering documentation for a design script in the database; and
> (Aleksic, especially: Fig.2, Item 46 and associated text)
>
> linking the documentation of the design script to the design elements comprising the design module.
> (Aleksic, especially: Fig.2, Item 46; Fig.4, Item 72; and associated text)

33. In regards to Claim 6, Claims 1 and 4 are rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

> 6. The method of claim 4, further comprising:
> entering documentation for the simulation elements in the database; and
> (Aleksic, especially: Fig.2, Item 46; Fig.5; and associated text)
>
> linking the documentation for the simulation elements with associated ones of the simulation elements.
> (Aleksic, especially: Fig.2, Item 46; Fig.4, Item 72; Fig.5; and associated text)

34. In regards to Claim 7, Claims 1, 4 and 6 are rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

> 7. The method of claim 6, further comprising:
> inspecting the functional design elements and simulation elements for associated documentation; and
> (Aleksic, especially: Fig.5 and associated text)

reporting documentation deficiencies in association with the functional design elements and simulation design elements.
(Aleksic, especially: Fig.2, Item 46; Fig.4, Item 72; Fig.5; and associated text)

35. In regards to Claim 8, Claim 1 is rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

8. The method of claim 1, further comprising:
inspecting the functional design elements for associated documentation; and
(Aleksic, especially: Fig.2, Item 46; Fig.4, Item 72; Fig.5; and associated text)

reporting documentation deficiencies in association with the functional design elements.
(Aleksic, especially: Fig.2, Item 46; Fig.4, Item 72; Fig.5; and associated text)

36. In regards to Claim 9, Claim 1 is rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

9. The method of claim 1, further comprising:
inspecting the functional design elements for undesirable design characteristics; and
(Aleksic, especially: Fig.5 and associated text)

reporting the undesirable design characteristics found in the functional design elements.
(Aleksic, especially: Fig.5 and associated text)

37. In regards to Claim 10, Claims 1 and 9 are rejected over Aleksic, as described above. In addition, Aleksic teaches the following limitations:

10. The method of claim 9, further comprising:
inspecting the functional design elements for undesirable hierarchical characteristics; and
(Aleksic, especially: Fig.3, Items 42, 49, 68; Fig.5 and associated text)

reporting discovered ones of the undesirable hierarchical characteristics.
(Aleksic, especially: Fig.3, Items 42, 49, 68; Fig.5 and associated text)

38. In regards to Claim 11, Claims 1 and 9 are rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

> 11. The method of claim 9, further comprising:
> inspecting the functional design elements for adherence to predefined design rules; and
> (Aleksic, especially: Fig.5 and associated text)
>
> reporting violations of the design rules.
> (Aleksic, especially: Fig.5 and associated text)

39. In regards to Claim 12, Claims 1, 9 and 11 are rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

> 12. The method of claim 11, further comprising providing assistance in specifying the design rules for the functional design elements
> (Aleksic, especially: Fig.5 and associated text)

40. In regards to Claim 13, Claims 1 and 9 are rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

> 13. The method of claim 9, further comprising:
> monitoring changes made to the functional design elements; and
> (Aleksic, especially: Fig.5 and associated text)
>
> indicating which of the functional design elements are dependent on the changes.
> (Aleksic, especially: Fig.5 and associated text)

41. In regards to Claim 14, Claim 1 is rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

> 14. The method of claim 1, further comprising:
> translating the functional design elements into a physical implementation; and
> (Aleksic, especially: Fig.5 and associated text)
>
> linking elements of the physical implementation with selected ones of the functional design elements.
> (Aleksic, especially: Fig.5 and associated text)

42. In regards to Claim 15, Claim 1 is rejected over Aleksic, as described above. In addition, Aleksic teaches the following limitations:

> 15. The method of claim 1, further comprising requiring specification of parameters at a top level of a hierarchy of the design module.
>
> (Aleksic, especially: Fig.3, Items 42, 49, 68; Fig.5 and associated text)

43. In regards to Claim 16, Claim 1 is rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

> 16. The method of claim 1, further comprising displaying the functional design elements linked to errors in the simulation results.
>
> (Aleksic, especially: Fig.5 and associated text)

44. In regards to Claim 17, Claims 1 and 16 are rejected over Aleksic as described above. In addition, Aleksic teaches the following limitations:

> 17. The method of claim 16, further comprising displaying documentation elements associated with errors in the simulation results.
>
> (Aleksic, especially: Fig.5 and associated text)

45. In regards to Claim 18, Aleksic teaches the following limitations:

> 18. An apparatus for developing a reusable electronic circuit design module, wherein the design module is comprised of one or more functional design elements comprising the design module, comprising:
>
> means for entering the functional design elements into a database;
>
> (Aleksic, especially: Fig.2, Item 38 and associated text)
>
> means for entering documentation elements into the database;
>
> (Aleksic, especially: Fig.2, Item 46 and associated text)
>
> means for linking the functional design elements with selected ones of the documentation elements;
>
> (Aleksic, especially: Fig.3, Item 36 and associated text)
>
> means for simulating a testbench with the design module, whereby simulation results are generated;
>
> (Aleksic, especially: Fig.5, Items 92-98 and associated text)

means for storing the simulation results in the. database; and
(Aleksic, especially: Fig.5, Items 92-102 and associated text)

means for linking the simulation results with the functional design elements.
(Aleksic, especially: Fig.3, Item 36, Fig.5, Items 92-102 and associated text)

46. In regards to Claim 19, Aleksic teaches the following limitations:

19. A system for developing a reusable electronic circuit design module, wherein the design module is comprised of one or more functional design elements comprising the design module, comprising:

a database arranged for storage of the design elements and documentation elements;
(Aleksic, especially: Fig.2, Items 38, 46 and associated text)

a design inspector coupled to the database, the design inspector configured and arranged to link the functional design elements with selected ones of the documentation elements;
(Aleksic, especially: Fig.3, Item 36 and associated text)

a debugging-support module coupled to the simulator and to the database, the debugging-support module configured and arranged to generate a netlist from the design module, wherein the netlist is suitable for simulation;
(Aleksic, especially: Fig.5, Items 92-98 and associated text)

a functional simulator coupled to the debugging-support module, the simulator configured and arranged to simulate a testbench with the design module, whereby simulation results are generated; and
(Aleksic, especially: Fig.5, Items 92-102 and associated text)

wherein the debugging-support module is further configured and arranged to store the simulation results in the database and link the simulation results with the functional design elements.
(Aleksic, especially: Fig.3, Item 36, Fig.5, Items 92-102 and associated text)

Claim Rejections - 35 USC § 103

47. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

48. The prior art used for these rejections is as follows:

49. Fields et al., U.S. Patent 6,223,326. (Henceforth referred to as "**Fields_1**").

50. Fields, C.A. "Creating Hierarchy in HDL-Based High Density FPGA Design". Proceedings EURO-DAC '95. Sept. 18-22, 1995. pp.594-599. (Henceforth referred to as "**Fields_2**").

51. Gentry. U.S. Patent 5,673,199. (Henceforth referred to as "**Gentry**").

52. Doughty, F. "6.111 Introductory Digital Systems Laboratory". Emacs Help page. Jan. 18, 2000. (Henceforth referred to as "**Emacs**").

53. "Intro. to Synopsys to XACT M1 Design Flow", Sept. 6, 1999. (Henceforth referred to as "**XACT**").

54. The claim rejections are hereby summarized for Applicant's convenience. The detailed rejections follow.

55. Claims 10 and 15 are rejected under 35 U.S.C. 103(a) as being obvious over Fields_1 in view of Fields_2.

The applied reference has a common assignee with the instant application. Based upon the earlier effective U.S. filing date of the reference, it constitutes prior art only under 35 U.S.C. 102(e). This rejection under 35 U.S.C. 103(a) might be overcome by: (1) a showing under 37 CFR 1.132 that any

invention disclosed but not claimed in the reference was derived from the inventor of this application and is thus not an invention "by another"; (2) a showing of a date of invention for the claimed subject matter of the application which corresponds to subject matter disclosed but not claimed in the reference, prior to the effective U.S. filing date of the reference under 37 CFR 1.131; or (3) an oath or declaration under 37 CFR 1.130 stating that the application and reference are currently owned by the same party and that the inventor named in the application is the prior inventor under 35 U.S.C. 104, together with a terminal disclaimer in accordance with 37 CFR 1.321(c). For applications filed on or after November 29, 1999, this rejection might also be overcome by showing that the subject matter of the reference and the claimed invention were, at the time the invention was made, owned by the same person or subject to an obligation of assignment to the same person. See MPEP § 706.02(l)(1) and § 706.02(l)(2).

56. **Claims 1, 4-12, and 15-19 are rejected under 35 U.S.C. 103(a) as being obvious over Gentry in view of Emacs.**

57. **Claims 2-3 and 13-14 are rejected under 35 U.S.C. 103(a) as being obvious over Gentry in view of Emacs and further in view of XACT.**

58. In regards to Claim 10, Claims 1 and 9 are rejected under 35 USC 102 over Fields_1, as described above. In addition, Fields_1 teaches the following limitations:

> 10. The method of claim 9, further comprising:
> inspecting the functional design elements for undesirable characteristics; and
>
> (Fields_1, especially: Fig.1, Item 106-110, and associated text)

reporting discovered ones of the undesirable characteristics.
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

However, Fields_1 does not expressly teach the inspection of undesirable hierarchical characteristics. Fields_2 does expressly teach this. (See Figures 1,2,4,7 and associated text).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Fields_1 with those of Fields_2 because "Re-structuring the design's hierarchy into mid-size modules (i.e., 100-200 CLBs) not only reduces the area utilization, but also allows the user to make small changes to the design and to easily locate logic for debugging of critical paths. This also reduces place and route execution times." (Fields_2, p.599, "Summary").

59. In regards to Claim 15, Claim 1 is rejected under 35 USC 102 over Fields_1, as described above. In addition, Fields_1 teaches the following limitations:

15. The method of claim 1, further comprising requiring specification of parameters at a top level of a hierarchy of the design module.
(Fields_1, especially: Fig.1, Item 106-110, and associated text)

However, Fields_1 does not expressly teach requiring hierarchical characteristics. Fields_2 does expressly teach this. (See Figures 1,2,4,7 and associated text).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Fields_1 with those of Fields_2 because "Re-structuring the design's hierarchy into mid-size modules (i.e., 100-

200 CLBs) not only reduces the area utilization, but also allows the user to make small changes to the design and to easily locate logic for debugging of critical paths. This also reduces place and route execution times." (Fields_2, p.599, "Summary").

60. In regards to Claim 1, Gentry teaches the following limitations:

> 1. A computer-implemented method for developing a reusable electronic circuit design module, wherein the design.module is comprised of one or more functional design elements comprising the design module, comprising:
>
> entering the functional design elements into a database;
>
> (Gentry, especially: Fig.2, Items 34, 36, 36', and associated text)
>
> simulating a testbench with the design module, whereby simulation results are generated;
>
> (Gentry, especially: Fig.2, Item 48, and associated text)
>
> storing the simulation results in the database; and
>
> (Gentry, especially: Fig.2, Items 48, 52, 30, 32, and associated text)
>
> linking the simulation results with the functional design elements.
>
> (Gentry, especially: Fig.4, Items 80, 82, 86, and associated text)

Gentry, however, does not expressly teach the following limitations:

> entering documentation elements into the database;
>
> linking the functional design elements with selected ones of the documentation elements;

On the other hand, the "Emacs" reference (pp.1-2), teaches that documentation, in the form of comments inserted into VHDL source files, was a standard part of the VHDL language. ("You are prompted for comments after object definitions (i.e. signals, variables, constants, ports) and after subprogram and process specifications if 'vhdl-prompt-for-comments' is non-nil.") The widely-

used Emacs text editor contains several functional commands for adding comment tags to a VHDL file, as described in the "Emacs" reference.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with Emacs because it was well known in the art that inserting comments into VHDL files was a standard feature of VHDL.

61. In regards to Claim 2, Claim 1 is rejected under 35 USC 103 over Gentry in view of Emacs, as described above. Gentry, however, does not expressly teach the following limitations:

> 2. The method of claim 1, further comprising:
> translating the functional design elements into a netlist; and
>
> linking elements of the netlist with selected ones of the functional design elements.

By applicant's own admission (p.6, line 34 to p.7, line 3), translating functional design elements into a physical implementation (a netlist, for example), by using DC2NCF, NGD2VHDL, and NGD2VER is old and well known in the art.

XACT teaches that NGD2VHDL and NGD2VER are used "to create a VHD or VER file that can be simulated for back annotation within Synopsys" (XACT, p.1). The term "back annotation" is interpreted as meaning that NGD2VHDL and NGD2VER are used to link the netlist with the functional design elements in Synopsys.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with XACT, because doing so to "to create a VHD or VER file that can be simulated for back annotation within Synopsys"

would enable keeping both versions updated whenever changes were made in one of the files.

62. In regards to Claim 3, Claim 2 is rejected under 35 USC 103 over Gentry in view of Emacs, as described above. Gentry, however, does not expressly teach the following limitations:

> 3. The method of claim 2, further comprising: translating the functional design elements into a physical implementation; and
>
> linking elements of the physical implementation with selected ones of the functional design elements.

By applicant's own admission (p.6, line 34 to p.7, line 3), translating functional design elements into a physical implementation (a netlist, for example), by using DC2NCF, NGD2VHDL, and NGD2VER is old and well known in the art.

XACT teaches that NGD2VHDL and NGD2VER are used "to create a VHD or VER file that can be simulated for back annotation within Synopsys" (XACT, p.1). The term "back annotation" is interpreted as meaning that NGD2VHDL and NGD2VER are used to link the netlist with the functional design elements in Synopsys.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with XACT, because doing so to "to create a VHD or VER file that can be simulated for back annotation within Synopsys" would enable keeping both versions updated whenever changes were made in one of the files.

63. In regards to Claim 4, Claim 1 is rejected under 35 USC 103 over Gentry in view of Emacs, as described above. Gentry also teaches the following limitations:

> 4. The method of claim 1, further comprising:
> entering simulation elements in the database; and
>
> (Gentry, especially: Fig.2, Items 48, 52, 30, 32, and associated text)
>
> linking the simulation elements to associated ones of the design elements.
>
> (Gentry, especially: Fig.4, Items 80, 82, 86, and associated text)

64. In regards to Claim 5, Claims 1 and 4 are rejected under 35 USC 103 over Gentry in view of Emacs, as described above. However, Gentry does not expressly teach the following limitations:

> 5. The method of claim 4, further comprising:
> entering documentation for a design script in the database; and
>
> linking the documentation of the design script to the design elements comprising the design module.

On the other hand, the "Emacs" reference (pp.1-2) teaches that documentation, in the form of comments inserted into VHDL source files, was a standard part of the VHDL language. The widely-used Emacs text editor contains several functional commands for adding comment tags to a VHDL file, as described in the "Emacs" reference.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with Emacs because it was well known in the art that inserting comments into VHDL files was a standard feature of VHDL.

65. In regards to Claim 6, Claims 1 and 4 are rejected under 35 USC 103 over Gentry in view of Emacs, as described above. However, Gentry does not expressly teach the following limitations:

> 6. The method of claim 4, further comprising:
> entering documentation for the simulation elements in the database; and
>
> linking the documentation for the simulation elements with associated ones of the simulation elements.

On the other hand, the "Emacs" reference (pp.1-2) teaches that documentation, in the form of comments inserted into VHDL source files, was a standard part of the VHDL language. The widely-used Emacs text editor contains several functional commands for adding comment tags to a VHDL file, as described in the "Emacs" reference.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with Emacs because it was well known in the art that inserting comments into VHDL files was a standard feature of VHDL.

66. In regards to Claim 7, Claims 1, 4 and 6 are rejected under 35 USC 103 over Gentry in view of Emacs, as described above. However, Gentry does not expressly teach the following limitations:

> 7. The method of claim 6, further comprising:
> inspecting the functional design elements and simulation elements for associated documentation; and
>
> reporting documentation deficiencies in association with the functional design elements and simulation design elements.

On the other hand, the "Emacs" reference (pp.1-2) teaches that documentation, in the form of comments inserted into VHDL source files, was a standard part of the VHDL language. The widely-used Emacs text editor contains several functional commands for adding comment tags to a VHDL file, as described in the "Emacs" reference.

Moreover, the "Emacs" reference teaches that "You are prompted for comments after object definitions (i.e. signals, variables, constants, ports) and after subprogram and process specifications if variable 'vhdl-prompt-for-comments' is non-nil." It is inherent that this functionality includes the claimed features of "inspecting the design elements for documentation", and "reporting documentation deficiencies in design elements."

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with Emacs because it was well known in the art that inserting comments into VHDL files was a standard feature of VHDL, and that it would be beneficial to check if documentation had indeed been entered.

67. In regards to Claim 8, Claim 1 is rejected under 35 USC 103 over Gentry in view of Emacs, as described above. However, Gentry does not expressly teach the following limitations:

> 8. The method of claim 1, further comprising:
> inspecting the functional design elements for associated documentation; and
>
> reporting documentation deficiencies in association

with the functional design elements.

On the other hand, the "Emacs" reference (pp.1-2) teaches that documentation, in the form of comments inserted into VHDL source files, was a standard part of the VHDL language. The widely-used Emacs text editor contains several functional commands for adding comment tags to a VHDL file, as described in the "Emacs" reference.

Moreover, the "Emacs" reference teaches that "You are prompted for comments after object definitions (i.e. signals, variables, constants, ports) and after subprogram and process specifications if variable 'vhdl-prompt-for-comments' is non-nil." It is inherent that this functionality includes the claimed features of "inspecting the design elements for documentation", and "reporting documentation deficiencies in design elements."

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with Emacs because it was well known in the art that inserting comments into VHDL files was a standard feature of VHDL, and that it would be beneficial to check if documentation had indeed been entered.

68. In regards to Claim 9, Claim 1 is rejected under 35 USC 103 over Gentry in view of Emacs, as described above. However, Gentry does not expressly teach the following limitations:

9. The method of claim 1, further comprising:

inspecting the functional design elements for

undesirable design characteristics; and

reporting the undesirable design characteristics found in the functional design elements.

On the other hand, the "Emacs" reference (pp.2 "Source File Compilation") teaches that "The syntax of the current buffer can be analyzed by calling a VHDL compiler". A VHDL compiler inherently inspects design elements for undesirable design characteristics, and reports them

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with "Emacs" because the use of a VHDL compiler to test for errors is integral to the use of VHDL.

69. In regards to Claim 10, Claims 1 and 9 are rejected under 35 USC 103 over Gentry in view of Emacs, as described above. However, Gentry does not expressly teach the following limitations:

10. The method of claim 9, further comprising:

inspecting the functional design elements for undesirable characteristics; and

reporting discovered ones of the undesirable characteristics.

On the other hand, the "Emacs" reference (pp.2 "Source File Compilation") teaches that "The syntax of the current buffer can be analyzed by calling a VHDL compiler". A VHDL compiler inherently inspects design elements for undesirable design characteristics, and reports them.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with "Emacs" because the use of a VHDL compiler to test for errors is integral to the use of VHDL.

70. In regards to Claim 11, Claims 1 and 9 are rejected under 35 USC 103 over Gentry in view of Emacs, as described above. However, Gentry does not expressly teach the following limitations:

> 11. The method of claim 9, further comprising:
> inspecting the functional design elements for adherence to predefined design rules; and
>
> reporting violations of the design rules.

On the other hand, the "Emacs" reference (pp.2 "Source File Compilation") teaches that "The syntax of the current buffer can be analyzed by calling a VHDL compiler". A VHDL compiler inherently inspects design elements for violations of design rules, and reports them.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with "Emacs" because the use of a VHDL compiler to test for errors is integral to the use of VHDL.

71. In regards to Claim 12, Claims 1, 9 and 11 are rejected under 35 USC 103 over Gentry in view of Emacs, as described above. However, Gentry does not expressly teach the following limitations:

> 12. The method of claim 11, further comprising providing assistance in specifying the design rules for the functional design elements

On the other hand, the "Emacs" reference (pp.2 "Source File Compilation") teaches that "The syntax of the current buffer can be analyzed by calling a VHDL compiler". A VHDL compiler, by inspecting design elements for violations of design rules, and reports them, thereby "provides assistance" in specifying the design rules.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with "Emacs" because the use of a VHDL compiler to test for errors is integral to the use of VHDL.

72. In regards to Claim 13, Claims 1 and 9 are rejected under 35 USC 103 over Gentry in view of Emacs, as described above. However, Gentry does not expressly teach the following limitations:

> 13. The method of claim 9, further comprising:
> monitoring changes made to the functional design elements; and
>
> indicating which of the functional design elements are dependent on the changes.

By applicant's own admission (p.6, line 34 to p.7, line 3), translating functional design elements into a physical implementation (a netlist, for example), by using DC2NCF, NGD2VHDL, and NGD2VER is old and well known in the art.

XACT teaches that NGD2VHDL and NGD2VER are used "to create a VHD or VER file that can be simulated for back annotation within Synopsys" (XACT, p.1). The term "back annotation" is interpreted as meaning that NGD2VHDL and NGD2VER are used to link the netlist with the functional design elements in Synopsys.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with XACT, because doing so to "to create a VHD or VER file that can be simulated for back annotation within Synopsys" would enable keeping both versions updated whenever changes were made in one of the files.

73. In regards to Claim 14, Claim 1 is rejected under 35 USC 103 over Gentry in view of Emacs, as described above. However, Gentry does not expressly teach the following limitations:

> 14. The method of claim 1, further comprising:
> translating the functional design elements into a physical implementation; and
>
> linking elements of the physical implementation with selected ones of the functional design elements.

By applicant's own admission (p.6, line 34 to p.7, line 3), translating functional design elements into a physical implementation (a netlist, for example), by using DC2NCF, NGD2VHDL, and NGD2VER is old and well known in the art.

XACT teaches that NGD2VHDL and NGD2VER are used "to create a VHD or VER file that can be simulated for back annotation within Synopsys" (XACT, p.1). The term "back annotation" is interpreted as meaning that NGD2VHDL and NGD2VER are used to link the netlist with the functional design elements in Synopsys.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with XACT, because doing so to "to create a VHD or VER file that can be simulated for back annotation within Synopsys"

would enable keeping both versions updated whenever changes were made in one of the files.

74. In regards to Claim 15, Claim 1 is rejected under 35 USC 103 over Gentry in view of Emacs, as described above. However, Gentry does not expressly teach the following limitations:

> 15. The method of claim 1, further comprising requiring specification of parameters at a top level of a hierarchy of the design module.

Moreover, the "Emacs" reference teaches that "You are prompted for comments after object definitions (i.e. signals, variables, constants, ports) and after subprogram and process specifications if variable 'vhdl-prompt-for-comments' is non-nil." It is inherent that this functionality includes the claimed features of "inspecting the design elements for documentation", and "reporting documentation deficiencies in design elements."

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with "Emacs" because the use of a VHDL compiler to test for errors is integral to the use of VHDL.

75. In regards to Claim 16, Claim 1 is rejected under 35 USC 103 over Gentry in view of Emacs, as described above. However, Gentry does not expressly teach the following limitations:

> 16. The method of claim 1, further comprising displaying the functional design elements linked to errors in the simulation results.

On the other hand, the "Emacs" reference (pp.2 "Source File Compilation") teaches that "The syntax of the current buffer can be analyzed by calling a VHDL compiler". A VHDL compiler, by inspecting design elements for errors, and reports them, thereby displaying the link between the elements and the errors they generate.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with "Emacs" because the use of a VHDL compiler to test for errors is integral to the use of VHDL.

76. In regards to Claim 17, Claims 1 and 16 are rejected under 35 USC 103 over Gentry in view of Emacs, as described above. However, Gentry does not expressly teach the following limitations:

> 17. The method of claim 16, further comprising displaying documentation elements associated with errors in the simulation results.

On the other hand, the "Emacs" reference (pp.2 "Source File Compilation") teaches that "The syntax of the current buffer can be analyzed by calling a VHDL compiler". A VHDL compiler, by inspecting design elements for errors, and reports them, thereby displaying the link between the elements and the errors they generate.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with "Emacs" because the use of a VHDL compiler to test for errors is integral to the use of VHDL.

77. In regards to Claim 18, Gentry teaches the following limitations:

18. An apparatus for developing a reusable electronic circuit design module, wherein the design module is comprised of one or more functional design elements comprising the design module, comprising:

means for entering the functional design elements into a database;

(Gentry, especially: Fig.2, Items 34, 36, 36', and associated text)

means for simulating a testbench with the design module, whereby simulation results are generated;

(Gentry, especially: Fig.2, Item 48, and associated text)

means for storing the simulation results in the. database; and

(Gentry, especially: Fig.2, Items 48, 52, 30, 32, and associated text)

means for linking the simulation results with the functional design elements.

(Gentry, especially: Fig.4, Items 80, 82, 86, and associated text)

Gentry, however, does not expressly teach the following limitations:

means for entering documentation elements into the database;

means for linking the functional design elements with selected ones of the documentation elements;

Emacs (pp.1-2), on the other hand, teaches that documentation, in the form of comments inserted into VHDL source files, was a standard part of the VHDL language. The widely-used Emacs text editor contains several functional commands for adding comment tags to a VHDL file, as described in the "Emacs" reference.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with Emacs because it was well known in the art that inserting comments into VHDL files was a standard feature of VHDL.

78. In regards to Claim 19, Gentry teaches the following limitations:

19. A system for developing a reusable electronic circuit design module, wherein the design module is comprised of one or more functional design elements comprising the design module, comprising:

a database arranged for storage of the design elements and documentation elements;

(Gentry, especially: Fig.2, Items 34, 36, 36', and associated text)

a debugging-support module coupled to the simulator and to the database, the debugging-support module configured and arranged to generate a netlist from the design module, wherein the netlist is suitable for simulation;

(Gentry, especially: Fig.2, Item 48, and associated text)

a functional simulator coupled to the debugging-support module, the simulator configured and arranged to simulate a testbench with the design module, whereby simulation results are generated; and

(Gentry, especially: Fig.2, Items 48, 52, 30, 32, and associated text)

wherein the debugging-support module is further configured and arranged to store the simulation results in the database and link the simulation results with the functional design elements.

(Gentry, especially: Fig.4, Items 80, 82, 86, and associated text)

Gentry, however, does not expressly teach the following limitations:

a database arranged for storage of the documentation elements;

a design inspector coupled to the database, the design inspector configured and arranged to link the functional design elements with selected ones of the documentation elements;

Emacs (pp.1-2), on the other hand, teaches that documentation, in the form of comments inserted into VHDL source files, was a standard part of the VHDL language. The widely-used Emacs text editor contains several functional commands for adding comment tags to a VHDL file, as described in the "Emacs" reference.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Gentry with Emacs because it was well known in the art that inserting comments into VHDL files was a standard feature of VHDL.

Conclusion

79. The following prior art, made of record and not relied upon, is considered pertinent to applicant's disclosure.
80. Naidu et al., U.S. Patent 5,752,002. Naidu teaches running a simulation of the circuit design, and storing the results in a log file, and linking the information (See Fig.3, “Address”) to functional design elements.
81. Devins et al., U.S. Patent 6,539,522. Devins teaches hierarchy in VHDL code, and in IP core reuse.
82. Poterie, B. “Storage Mechanism for VHDL Intermediate Form”, Proceedings of the European Design Automation Conference (EDAC), 1990. March 12-15, 1990. pp.506-510. Poterie teaches a a database for storing VHDL code.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ayal I. Sharon whose telephone number is (703) 306-0297. The examiner can normally be reached on Monday through Thursday, and the first Friday of a biweek, 8:30 am – 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kevin Teska can be reached on (703) 305-9704. Any response to this office action should be mailed to:

Director of Patents and Trademarks
Washington, DC 20231

Hand-delivered responses should be brought to the following office:

4th floor receptionist's office
Crystal Park 2
2121 Crystal Drive
Arlington, VA

The fax phone numbers for the organization where this application or proceeding is assigned are:

Official communications:	(703) 746-7239
Non-Official / Draft communications	(703) 746-7240
After Final communications	(703) 746-7238

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist, whose telephone number is: (703) 305-3900.

Ayal I. Sharon

Art Unit 2123

May 7, 2003

SBroda

SAMUEL BRODA, ESQ.
PRIMARY EXAMINER